Scientia Iranica E (2015) 22(3), 1203–1217

Sharif University of Technology
Scientia Iranica
Transactions E: Industrial Engineering
www.scientiairanica.com

An empirical comparison of simulated annealing and iterated local search for the hierarchical single allocation hub median location problem

M.H. Fazel Zarandi[a,*], **S. Davari**[b] **and S.A. Haddad Sisakht**[c]

a. *Department of Industrial Engineering, Amir Kabir University of Technology, 424 Hafez Avenue, Tehran, Iran.*
b. *Department of Manufacturing Systems and Industrial Engineering, Sabanci University, Orhanli-Tuzla, 34956 Istanbul, Turkey.*
c. *Department of Industrial and Manufacturing Systems Engineering, Iowa State University, 50011 Ames, USA.*

Received 12 June 2013; received in revised form 18 February 2014; accepted 29 September 2014

KEYWORDS
Location;
Simulated annealing;
Iterated local search;
Heuristics;
Metaheuristics.

Abstract. The Hub Location Problem (HLP) has been an attractive area of research for more than four decades. A recently proposed problem in the area of hub location is the hierarchical Single-Allocation Hub Median Problem (SA-H-MP), which is associated with finding the location of a number of hubs and central hubs, so that the total routing cost is minimized. Owing to the problem's complexity and intractability, this paper puts forward two metaheuristics, Simulated Annealing (SA) and Iterated Local Search (ILS), and compares their performances. Results show that while both algorithms are able to reach optimal solutions on the standard CAB dataset, their runtimes are negligible and considerably lower compared to the runtimes of exact methods.

1. Introduction

Facility Location (FL) concerns the choice of the location of one or multiple facilities in a given geographical space and subject to some constraints to optimally fulfill predetermined objectives [1]. The problem of locating facilities is not new to the operations research community. The challenge of where to best site facilities has inspired a rich, colorful and ever growing body of literature [2]. Models of FL have been studied as discrete, network, and continuous models, and employed in a variety of applications, such as carriers, airports, supply chains, and service centers, etc. For the sake of brevity, this paper does not provide a comprehensive introduction of FL.

From the late 60s, the Hub Location Problem (HLP) turned out to be an intricate, yet noteworthy and practical variant of facility location. One indication for its increasing relevance and fascination is the large number of publications dealing with different types of hub location problems over the last 10 years [3]. Hub-and-spoke networks are used in telecommunications, the airline industry, supply chain management, and in many other contexts. The p-hub median can be considered a practical model to establish facilities on a network to reach lower costs and higher service levels. The objective of the p-hub median problem is to minimize the total transportation cost (time, distance, etc.) needed to serve the given set of flows, given n demand nodes, flow between origin-destination pairs, and the number of hubs to locate [4].

Design of hierarchical hub networks is a variant of traditional HLP, which is applicable in many kinds of networks. A sample hierarchical hub network is depicted in Figure 1 where circles, rectangles, and hexagons depict demand nodes, non-central hubs, and central hubs, respectively. Prominent applications of the problem are in cross-docking and distribution network design. The hierarchical hub network design

*. *Corresponding author. Tel.: +98 21 64545378*
E-mail address: zarandi@aut.ac.ir (M.H. Fazel Zarandi)

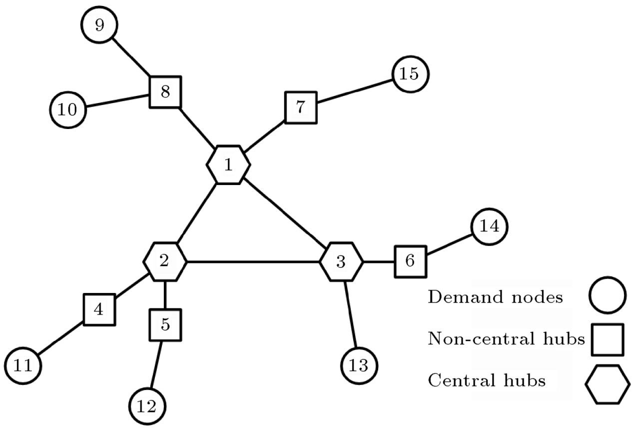

Figure 1. A three level hub-and-spoke network with 3 central hubs, 8 hubs, and 15 nodes.

has been addressed by a number of scholars around the world. One of the most recent publications in hierarchical HLP is [5], which presents the model for a three-level hub network with single assignment. Although Yaman [5] is a great step forward in modeling hierarchical hub location problems, it can be argued that its solution needs a great deal of time, which makes using exact methods impossible for larger problems. Therefore, this paper presents efficient SA and ILS procedures, which are capable of reaching solutions in considerably less time.

Before going to the next section of the paper, we clarify the main contribution of this paper. To the best of our knowledge, no heuristic or metaheuristic has been proposed for the hierarchical Single-Allocation Hub Median Problem (SA-H-MP) to date. Therefore, the main contribution of this paper is the proposal of efficient SA and ILS procedures to solve SA-H-MP, and to address the question of whether SA or ILS outperforms the other for SA-H-MP. To do so, both algorithms are designed in such a way as to ensure a fair comparison, and the two procedures have been evaluated from different points of view.

The outline of this paper is as follows: The paper proceeds with a concise literature review of HLP. Section 3 gives the definition of the problem and its mathematical formulation. In Section 4, the proposed solution procedures are presented and their modules are discussed in detail. Numerical experiments and in-depth analyses of results appear in Section 5. Finally, conclusions and the outlook for potential future research are highlighted in Section 6.

2. Literature review

As already stated, the concept of HLP dates back to the 1960s, when Goldman [6] defined the network hub location problem for the first time. Later, the study of HLP flourished due to the influential paper of O'Kelly [7] presenting the first mathematical formulation for a HLP. Campbell [8] is considered to be the first to propose a mathematical formulation for the single-allocation hub median problem. Many variations to the model of Campbell [8] are present in the literature, such as Skorin-Kapov et al. [9] and Ebery [10]. In this paper, we do not aim to review all pertinent work on hub-median location problems, instead we refer interested readers to the valuable review by Alumur and Kara [4].

The hierarchical location problem has been addressed in the literature before. A review of these papers before 2007 is available in Sahin and Sural [11]. Other papers in this field after 2007 include Alumur et al. [12], Lee and Lee [13], and Davari et al. [14].

In the last decade, some contributions have been made to hub location literature, some influential publications of which are cited in Table 1.

Using non-exact methods to solve HLP has gained some attention among researchers, such as the Tabu search by Calik et al. [15], the genetic algorithm by Cunha and Silva [16], Tabu search by Skorin-Kapov and Skorin-Kapov [17], the ant colony by Meyer et al. [18], simulated annealing by Abdinnour-Helm [19], and the variable neighborhood search by Ilic et al. [20].

From the above, it becomes clear that the hierarchical HLP is still a relatively unexplored problem. Moreover, considering the review paper by Alumur and Kara [4] and our extensive review of the publications from 2008 to date, it becomes clear that non-exact methods have not greatly attracted the attention of scientists to date. Hence, this paper contributes to the literature by the proposal of efficient SA and ILS procedures in solving SA-H-MP.

3. Problem definition

The hierarchical hub median problem is addressed in Yaman [5], presenting a mixed integer model when the triangle inequality holds about the costs in the network. He assumed that there is a three-level hub network, where the top level, which connects the central hubs, is a complete network, and the second and third levels are star networks connecting each hub to a central hub and each demand node to one and only one hub or central hub. In such a network, each flow from an origin to a destination may visit up to four hub nodes, as the flow from node 9 to 14 in Figure 1, which is comprised of two non-central hubs and two central hubs. In other words, owing to the special types of connection in a complete hierarchical hub network, no connection with more than four hubs is possible. In their formulation, I, H, and C represent the set of nodes, possible locations of hub nodes, and possible locations of central hub nodes, respectively. It should be noted that $H \subseteq I$ and $C \subseteq H$. Z_{ijl} is a binary variable taking a one value if node $i \in I$ is assigned to hub $j \in H$, and hub

Table 1. Some major extensions to the classical hub location problem in the last decade.

Subject	**Author(s)**
Conditional p-hub median location problem	Eiselt and Marianov [21]
Aggregation in HLP	Gavriliouk [22]
Stochastic p-hub center	Sim et al. [23]
Reliable hub location problem	Kim and O'Kelly [24]
HLP for time definite transportation	Campbell [25]
Game theoretical model in HLP	Lin and Lee [26]
HLP with multiple capacity levels	Correia et al. [27]
Hierarchical HLP network for dual express services	Lin [28]
Decentralized management in HLP	Vasconcelos et al. [29]
HLP with balancing requirements	Correia et al. [30]
Stochastic uncapacitated HLP	Contreras et al. [31]
Partitioning-hub-location-routing problem	Catanzaro et al. [32]
Allocation strategies in HLP	Yaman [33]
Ordered median hub location problem	Puerto et al. [34]
Hub covering with backup coverage	Fazel Zarandi et al. [35]
Hub location with imprecise demand locations	Davari et al. [36]
Competitive hub location and pricing	Luer-Villagra and Marianov [37]
Many-to-many hub location-routing	De Camargo et al. [38]

j is allocated to the central hub, $l \in C$. It is worth mentioning that if $j \in H$ becomes a hub node and allocated to the central hub, $l \in C$, then the value of Z_{jjl} equals 1. Moreover, if node $l \in C$ is a hub, then, variable Z_{lll} takes a value of 1. g^i_{jl} is the amount of flow, which has node $i \in I$ as the origin or destination travelling between hub $j \in H$ and central hub $l \in C$. f^i_{kl} denotes the amount of traffic of node $i \in I$ as the origin travelling from central hub $k \in C$ to central hub $l \in C \backslash \{k\}$. Moreover, p and p_0 are the number of hubs and central hubs to be located. The amount of traffic to be routed from node $i \in I$ to node $m \in I$ is shown as t_{im}. The cost of routing a unit of flow from node $i \in I$ to node $j \in I$ is shown as d_{ij} ($d_{ii} = 0$ and $d_{ij} = d_{ji}$ for all pairs of i and j). The discount factor of routing between hubs and central hubs is shown as α_H, and the discount factor of routing between central hubs as α_C. The mathematical formulation is as follows:

$$\min \sum_{i\in I}\sum_{m\in I}(t_{im}+t_{mi})\sum_{j\in H} d_{ij}\sum_{l\in C} z_{ijl} + \sum_{i\in I}\sum_{j\in H}\sum_{l\in C\backslash\{j\}} \alpha_H d_{jl} g^i_{jl} + \sum_{i\in I}\sum_{j\in C}\sum_{l\in C\backslash\{j\}} \alpha_C d_{jl} f^i_{jl}, \tag{1}$$

$$\sum_{j\in H}\sum_{l\in C} z_{ijl} = 1 \quad \forall i \in I, \tag{2}$$

$$z_{ijl} \le z_{jjl} \quad \forall i \in I,\ j \in H \setminus \{i\},\ l \in C, \tag{3}$$

$$\sum_{m\in H} z_{jml} \le z_{lll} \quad \forall j \in H,\ l \in C \setminus \{j\}, \tag{4}$$

$$\sum_{j\in H}\sum_{l\in C} z_{jjl} = p, \tag{5}$$

$$\sum_{l\in C} z_{lll} = p_0, \tag{6}$$

$$\sum_{k\in C\backslash\{l\}} f^i_{lk} - \sum_{k\in C\backslash\{l\}} f^i_{kl} = \sum_{m\in I} t_{im}\sum_{j\in H}(z_{ijl} - z_{mjl}) \quad \forall i \in I,\ \ l \in C, \tag{7}$$

$$g^i_{jl} \ge \sum_{m\in I\backslash\{j\}} (t_{im}+t_{mi})(z_{ijl} - z_{mjl}) \quad \forall i \in I,\ j \in H,\ l \in C \setminus \{j\}, \tag{8}$$

$$z_{ljl} = 0 \quad \forall j \in H,\ l \in C \setminus \{j\}, \tag{9}$$

$$g^i_{jl} \ge 0 \quad \forall i \in I,\ j \in H,\ l \in C, \tag{10}$$

$$f^i_{kl} \ge 0 \quad \forall i \in I,\ k \in C,\ l \in C \setminus \{k\}, \tag{11}$$

$$z_{ijl} \in \{0,1\} \quad \forall i \in I,\ j \in H,\ l \in C. \tag{12}$$

Eq. (1) computes the fitness of solutions, which is the sum of routing between demand nodes and allocated hubs/central hubs, between the hub nodes and central hubs, and also between the central hubs. Constraint (2) states that each demand node must be assigned to one, and only one hub and a central hub. Constraint (3) is used to guarantee that when node i is assigned to hub j and central hub l, the jth node must be a hub that is allocated to a central hub at node l. Constraint (4) governs that if node j is assigned to the central hub, l, then, node l must be a central hub. Constraints (5) and (6) determine the number of hubs and central hubs to be located. Constraint (7) is the traditional flow balance constraints which are customized for use in SA-H-MP. Constraints (8) and (10) are used in order to compute the values of g^i_{jl} in terms of the assignment variables. Equation (9) strengthens the LP relaxation of the problem. Constraints (11) and (12) are ordinary constraints to limit z values to take binary values, and f values to be positive.

Yaman [5] proved that the SA-H-MP is NP-hard. This means that for comparatively larger datasets, exact solution approaches are handicapped to reach optimal solutions in a reasonable amount of time. This paper aims to propose efficient SA and ILS procedures to solve SA-H-MP that is capable of reaching optimal solutions in negligible time.

4. Solution procedure

In what follows, the components of the proposed procedures are elaborated.

4.1. Simulated annealing

Simulated Annealing (SA) is a stochastic, renowned and popular metaheuristic with a well-known lineage. It was first introduced by Kirkpatrick et al. [39] and has since been widely used to solve many real-world combinatorial problems. Multiple characteristics of SA have made it a favorite metaheuristic since its introduction, such as its convergence properties, ease of implementation, and analogy to nature. Recently, SA has gained broad appeal to solve various problems, such as location-routing problems [40], job shop scheduling [41], image processing [42], and even HLP [43].

There are some earlier attempts to use simulated annealing to solve variants of HLP. Rodríguez-Martín and Salazar-González [44] presented a simulated annealing approach to solve the capacitated version of HLP, modeling the problem as an $M/M/1$ queuing system. The paper by Ernst and Krishnamoorthy [45] is one of the first attempts to employ SA in HLP literature. They used the SA upper bound to develop a branch-and-bound solution procedure. A hybrid approach using a genetic algorithm and simulated annealing has been presented in [46] to solve a capacitated single-allocation hub covering location problem. Ernst and Krishnamoorthy [47] developed a hybrid heuristic of SA and random descent. Chen [48] presented a hybrid heuristic based on SA and the Tabu list. It should be noted that a comparison of our procedure with earlier ones is impossible, since they address different problems.

4.1.1. Initialization

Initialization of solutions plays a fundamental role in the success of any local search metaheuristic. To devise a rational initialization procedure for the problem of this paper, a set of preliminary experiments were carried out. Based on the results, we drew a tentative conclusion that nodes which send and receive higher volumes of flow are usually present in the set of hubs. For instance, in the CAB dataset, the flow to nodes 3, 4, 12, 14, 17 and 25 comprise more than 50 percent of the total flow. To employ this finding, the total incoming flow to each node is found and then, using the roulette-wheel selection, hubs are selected till the number of pre-determined hubs is located. It should be pointed out that the set of central hubs are randomly selected within the set of hubs. Then, using the nearest neighbor rule, each demand node is allocated to its nearest hub node. Take note that the allocated hub could be a hub or a central hub. Then, the current solution is improved for a number of iterations using each of the procedures (this will be elaborated on later in this paper). Figure 2 shows a sample initial solution, in which demand nodes are allocated to their nearest neighbors and the solution has been improved using a pre-determined set of moves. One may pose a question regarding the applicability of this procedure for other datasets. Our review of other hub-related datasets and some non-hub-related datasets showed that, normally, incoming flows to nodes follow a similar pattern. As an example, in the Turkish network dataset, comprised of 81 nodes, 8 nodes account for nearly 40 percent of the flow.

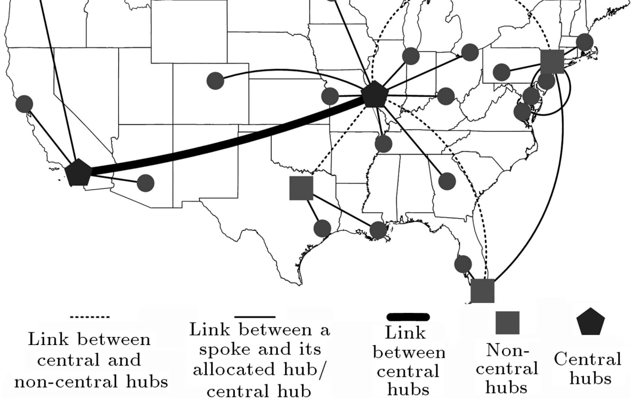

Figure 2. A sample improved initial solution.

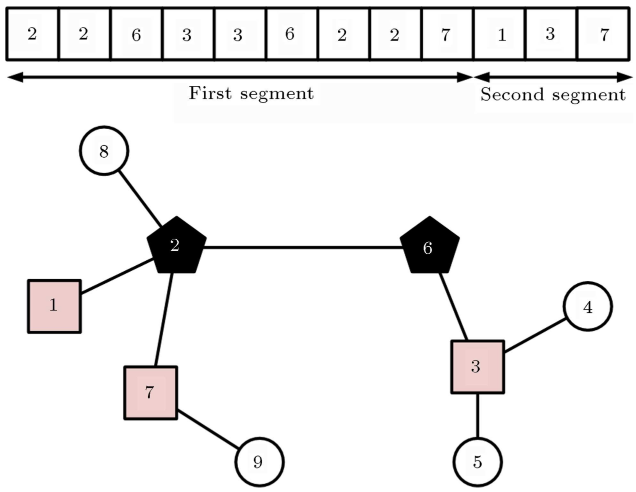

Figure 3. A sample encoding scheme and its equivalent solution representation.

4.1.2. Encoding scheme

An effective encoding scheme has a significant impact on the performance of SA. Although there are plenty of methods to encode the solutions of SA-H-MP, this paper uses an efficient scheme, which is comprised of two segments, each containing a set of indices. The value in each element of the first segment shows the index of the hub/central hub to which the node is allocated. In addition, the second segment contains the indices of hub nodes, $i \in H \setminus C$. Clearly, the set of central hubs is easily determined by finding those nodes which are not allocated to any other node in the first segment. Assuming the cardinality of nodes, hub nodes, and central hubs to be n, h, and c, respectively, while the length of the first segment equals n, the second segment contains h bits. Therefore, the length of the solution string is always fixed and equals $n + h$. Figure 3 shows an example which could be used to further explain the encoding scheme. It should be recalled that the direct allocation of a spoke to a central hub is possible, as shown for node 8.

4.1.3. Termination criterion

Our preliminary experiments showed that running the procedure for more than 150 seconds rarely improves the solution quality. Therefore, the procedure is run for 150 seconds, regardless of the problem size or its parameters.

4.1.4. Neighborhood Search Structure (NSS)

In order to search for better solutions, we define the set $N_i(X)$ to be the set of solutions neighboring a solution, X, using the ith mechanism. The first type of neighborhood search, $N_1(X)$, deals with substitution of the role of node $i \in H \setminus C$, and another node, $j \in C$. The second move, $N_2(X)$, is carried out to change the set of hub nodes by removing node $i \in H$ and adding another node, $j \in I \setminus H$, to the list of hub nodes. It should be pointed out that the hub to be removed can be a hub or a central hub, due to the fact that $C \subseteq H$. Another possible move is associated with trying to improve the solution by changing the allocation of node $i \in I \setminus C$ to node $j \in H$. The third move, $N_3(X)$, deals with carrying out this move for all the nodes in $I \setminus C$. To put it in simpler terms, N_3 modifies the allocation of one demand node at a time without changing the set of located hubs, and performs this change for a number of iterations, like π iterations. Figure 4 shows a sample solution and the performance of the three proposed moves on the solution representation. In this figure, the changes to the solution are shown by highlighting that specific bit of the solution string.

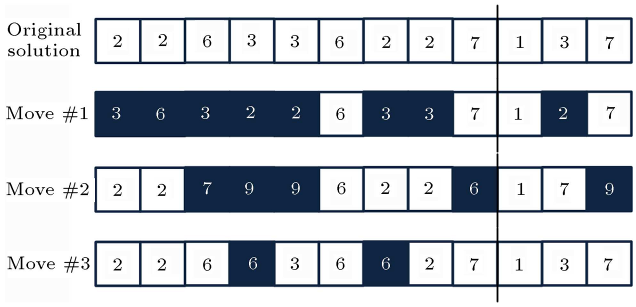

Figure 4. The performance of the three moves.

To go through the performance of these moves, several experiments were conducted to see if these moves have any significant role in improving the solution. Twenty combinations of move chances were generated using intuition and the problems were run five times using each combination of move chances. To run experiments, we deliberately selected one of the most difficult instances of test problems, which is the case with $p = 6$, $p_0 = 4$, and $\alpha_C = \alpha_H = 0.6$. Results of this analysis are shown in Figure 5 as a box-plot diagram. Results give clear indication that scenario #12 is the best scenario to be used, since it reaches optimal solutions in four out of five times. Moreover, the worse performance of this move has an error of about 1%. Thus, the 12th scenario has been used in

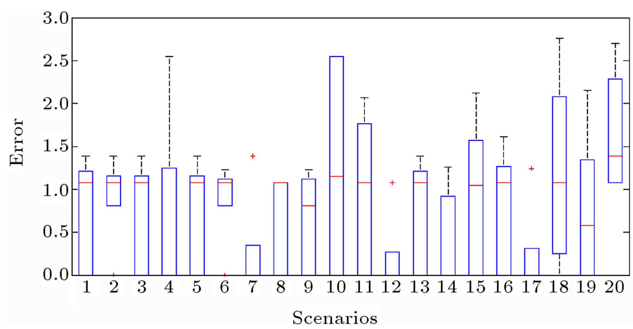

Figure 5. Comparing the scenarios of move chances for the problem with $\alpha_C = \alpha_H = 0.6$, $p = 6$, and $p_0 = 4$.

which first, second, and third moves have chances equal to 50, 30 and 20 percent.

4.1.5. Cooling schedule
Obviously, SA is quite sensitive to its cooling schedule. Therefore, great care should be taken in defining an effective and efficient cooling schedule. A common rule in defining a cooling schedule is to decrease the temperature neither too slowly nor too fast to prevent inefficient and unstable performance.

In this paper, three types of cooling schedule are used as follows. Further details are obtainable in Lundy and Mees [49]:

- Linear cooling rate: $T_l = T_0 - l\frac{T_0 - T_f}{N}$; $l = 1, 2, ..., N$;
- Exponential cooling rate: $T_l = \frac{A}{l+1} + B$; $A = \frac{(T_0 - T_f)(N+1)}{N}$; $B = T_0 - A$; $l = 1, 2, ..., N$;
- Hyperbolic: $T_l = \frac{1}{2}\left(T_0 - T_f)(1 - tgh(\frac{10l}{N} - 5)\right) + T_f$; $l = 1, 2, ..., N$.

In these equations, T_0, T_f, and T_l represent initial temperature, stopping temperature, and temperature of iteration l, respectively. Moreover, N is the number of temperatures between T_0 and T_f, and tgh is the tangent hyperbolic function.

Another important parameter of SA is its initial temperature, which is of great importance. To set the initial temperature, we have employed the same procedure as Dong et al. [50], which is as follows. Given the initial temperature, T_0, a sufficient number of solutions are generated in which a solutions are improving solutions and b solutions lead to worse fitness values. Then, the initial probability of acceptance is found using Eq. (13):

$$\chi = \frac{a + b\left[1 - (1-h)\overline{\Delta f}/T_0\right]^{1/(1-h)}}{a + b}, \tag{13}$$

where $a = k * b$, and $\overline{\Delta} f$ is the average worsening of the fitness value in b iterations. Now, the initial temperature is found as Eq. (14).

$$T_0 = \frac{1-h}{1 - \left[(k+1)\chi - k\right]^{1-h}} * \overline{\Delta f}. \tag{14}$$

4.1.6. Parameter setting and calibration
To know whether differences in the results obtained using various parameters is significant or not, we have conducted an ANOVA test. To do so, initialization type, cooling schedule, stall limit, the value of h, and the value of χ were considered. We ran five replicates of each experiment and the results were analyzed by means of a multi-factor analysis of variance (ANOVA). To assess the performance of the proposed initialization, three kinds of initialization type are used as: (I) locating hubs/central hubs and allocation of demand nodes randomly, (II) locating hubs/central hubs randomly, but allocating demand nodes using the nearest neighbor rule and improving the solution for a number of runs, (III) locating hubs and central hubs based on RWS, allocated based on the nearest neighbor rule, and improving the solution for a number of runs.

Different levels of the parameters used in the ANOVA test are given in Table 2 with $3*3*2*2*2*5 = 360$ runs to be carried out in total. Results in Table 3 testify that the differences in results are significant using various cooling schedules at a 99% confidence level. While the interaction of (cooling schedule, χ) is significant at a 95% level, the interaction of (stall limit, initialization type) is significant at a 90% confidence

Table 2. Levels used for solution parameters.

Factor	Symbol	Number of levels	Levels
Cooling schedule	A	3	$A(1)$-linear $A(2)$-exponential $A(3)$-hyperbolic
χ	B	2	$B(1)$-0.7 $B(2)$-0.9
Stall limit	C	2	$C(1)$-60 $C(2)$-150
H	D	2	$D(1)$-0.8 $D(2)$-0.9
Initial solution generation	E	3	$E(1)$-initialization I $E(2)$- initialization II $E(3)$- initialization III

Table 3. The results of ANOVA test.

Source	Sum Sq	d.f.	Mean Sq	F	Prob > F
A	1.58e+18	2	7.92e+17	51.06	**0****
B	2.54e+16	1	2.54e+16	1.64	0.2019
C	2.99e+13	1	2.99e+13	0	0.965
D	5.52e+12	1	5.52e+12	0	0.985
E	5.00e+16	2	2.50e+16	1.61	0.2008
$A*B$	1.09e+17	2	5.43e+16	3.5	**0.0313****
$A*C$	3.99e+15	2	2.00e+15	0.13	0.8792
$A*D$	8.58e+16	2	4.29e+16	2.77	0.0644
$A*E$	1.92e+17	4	4.79e+16	3.09	**0.0161****
$B*C$	3.49e+15	1	3.49e+15	0.23	0.6354
$B*D$	2.30e+15	1	2.30e+15	0.15	0.7001
$B*E$	4.53e+16	2	2.26e+16	1.46	0.2337
$C*D$	2.99e+16	1	2.99e+16	1.93	0.1658
$C*E$	8.60e+16	2	4.30e+16	2.77	**0.0639****
$D*E$	2.20e+16	2	1.10e+16	0.71	0.4925
Error	5.16e+18	333	1.55e+16		
Total	7.40e+18	359			

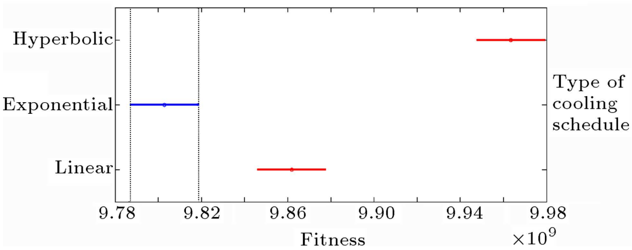

Figure 6. The Scheffé test to compare the types of cooling schedule.

level. This finding is of great importance and helps to fine-tune the algorithm to reach better solutions.

Knowing the significance of some parameters, an investigation into finding the optimal values of these parameters is advisable. This was carried out using the Scheffé multiple comparison test. A sample output of the analysis is given in Figure 6, which clearly shows that there are substantial differences between the three cooling schedules. Moreover, the exponential cooling type is superior to the other two, in terms of the solution quality. The same analysis was carried out for the other parameters and the optimal levels were used in the computational experiments as follows: Exponential cooling type; 150 iterations as the stall limit; $\chi = 0.7$; and the third initialization type.

4.2. Iterated local search

ILS is a neighborhood exploration procedure that was introduced by Lourenco et al. [51]. It has been applied to a wide range of applications, including quadratic assignment problems [52], and vehicle routing problems [53].

The procedure initiates from an initial solution, s_0, which should be improved using local search mechanisms. In each iteration, a perturbation is applied to obtain a new solution, s', and the local optimum around s' is found, which is shown as s''. The current solution is replaced by the new solution, s'', if some acceptance criteria are met. The same procedure is pursued until some termination criteria are satisfied. It is to be pointed out that although the tuning of the proposed ILS is not elaborated upon here, due to space limitations, the optimal values are reported in the following sections.

4.2.1. Initialization and encoding scheme

The same initialization types and encoding scheme as the proposed SA have been used here. Moreover, the re-initialization mechanism is applied whenever ILS fails to improve the solution for n_{stall} consecutive iterations. In this study, n_{stall} is tuned to equal 1000 iterations for the proposed ILS.

4.2.2. Perturbation

An efficient ILS needs a proper diversification procedure to explore the search space. The importance of the perturbation mechanism is obvious. On the one hand, a perturbation that is not strong enough might not enable the algorithm to escape from the basin of attraction of the current solution. On the other hand, a perturbation that is too strong would make the algorithm similar to a random restart local search [54].

In the proposed ILS, in order to be resourceful in perturbing solutions, $N_1(X)$ and $N_2(X)$ are used as perturbation moves. In the perturbation phase of the proposed ILS, if no better solution is found after perturbing the solution using $N_1(X)$ and its subsequent local search, $N_2(X)$ is applied to explore the search space.

4.2.3. Local search

A local search is a procedure to find a local optimum by exploring the neighborhood of a solution. Due to the characteristics of the problem at hand, only one type of local search is applied here, which is identical to $N_3(X)$ already used in the SA. It is to be noticed that two kinds of policy for applying the local search procedure are examined in this paper. In the first policy, once a better solution is found, it is applied immediately without looking for any other solution. However, the second policy is associated with selection of the best solution among all the solutions tested in the neighborhood of s'. Our preliminary experiments showed that using the second policy leads to better solutions. Therefore, in each iteration of the ILS, the best solution in the neighborhood of the current solution is selected.

4.2.4. Acceptance criterion

Two possible approaches to accepting solutions are to accept better solutions only, or to follow a similar approach to SA by accepting worse solutions with non-increasing probability. Although these two could be efficient in many other problems, our preliminary experiments clearly show that a simpler approach outperforms the above-mentioned methods. This method is as follows: If a solution has a cost less than $f(s^*)(1+\delta)$, it will replace the current solution, where s^* is the best solution so far, and δ is a user-defined positive parameter that has considerable impact on ILS performance. The value of δ is tuned in our study to equal 0.001.

4.2.5. Termination criterion

In order to compare SA and ILS fairly, the termination criterion of the proposed ILS is to run for 150 seconds.

5. Numerical examples

5.1. Test problems

In this section, numerical examples are given and analyses of the problem parameters are provided. In order to examine the performance of the proposed solution algorithm, the standard CAB dataset has been used. The time needed to solve problem instances to optimality exceeds 10 hours in some cases. Therefore, we believe that the standard CAB dataset is suitable for targeting in this paper, in order to compare the performance of SA and ILS. Thus, larger datasets have not been considered in this paper. Moreover, all 25 cities in the CAB dataset are considered to be candidate nodes to establish hubs, i.e. $H = C = I$. Overall, five sets of test problem are run for $(\alpha_C, \alpha_H) = (0.6, 0.6)$ as case I, $(\alpha_C, \alpha_H) = (0.8, 0.8)$ as case II, $(\alpha_C, \alpha_H) = (0.9, 0.9)$ as case III, $(\alpha_C, \alpha_H) = (0.6, 0.9)$ as case IV, and $(\alpha_C, \alpha_H) = (0.8, 0.9)$ as case V. For each of these parameter pairs, twelve test problems for various values of p and p_0 with up to 6 hub nodes are considered. The performance measure used to evaluate the efficiency of the proposed SA is the Relative Percentage Deviation (RPD), which is computed as follows:

$$\text{Relative Percentage Deviation (RPD)} = \frac{\text{Fitness}_{\text{SA/LLS}} - \text{Fitness}_{\text{CPLEX}}}{\text{Fitness}_{\text{CPLEX}}} * 100, \tag{15}$$

where $\text{Fitness}_{\text{CPLEX}}$ and $\text{Fitness}_{\text{SA/ILS}}$ are the results obtained from CPLEX as the optimal solution and from SA/ILS, respectively. It should be pointed out that in this paper, instead of reporting the best solution found, we report the worst, average, and best results to represent the performance of the proposed SA in a better way.

5.2. Computer specifications

All the test problems were run on a 2 GHz CPU equipped with 2 Gigabytes of RAM, using the CPLEX 12.2 solver. Moreover, it should be noted that C++ programming language was used to code the meta-heuristics.

5.3. Results, validation and discussions

To compare the performance of the proposed SA with exact solution methods, a set of experiments were carried out. As already stated, overall, there were sixty instances to be solved. Each of these instances was solved five times and the best, average, and worst errors are reported in Tables 4 to 9. Please take

Table 4. Results of CPLEX and the proposed SA for $(\alpha_C, \alpha_H) = (0.6, 0.6)$ and $(\alpha_C, \alpha_H) = (0.8, 0.8)$.

(α_C, α_H)	p	p_0	CPLEX Time (s)	CPLEX Optimal	SA Time (s)	Best BFS	Best Gap%	Average Fitness	Average Gap%	Worst Fitness	Worst Gap%
(0.6,0.6)	3	1	114	8,840,639,236	150	8,840,639,236	0.00%	8,840,639,236	0.00%	8,840,639,236	0.00%
	3	2	812	8,840,639,236	150	8,840,639,236	0.00%	8,840,639,236	0.00%	8,840,639,236	0.00%
	3	3	10	8,826,647,392	150	8,826,647,392	0.00%	8,826,647,392	0.00%	8,826,647,392	0.00%
	4	**2**	**2296**	**8,366,925,044**	**150**	**8,366,925,044**	**0.00%**	**8,377,034,188**	**0.12%**	**8,379,561,473**	**0.15%**
	4	3	1340	8,105,406,729	150	8,105,406,729	0.00%	8,105,406,729	0.00%	8,105,406,729	0.00%
	4	4	13	8,020,821,500	150	8,020,821,500	0.00%	8,020,821,500	0.00%	8,020,821,500	0.00%
	5	**3**	**4505**	**7,792,523,096**	**150**	**7,817,662,911**	**0.32%**	**7,848,129,867**	**0.71%**	**7,904,113,874**	**1.43%**
	5	**4**	**3576**	**7,649,779,782**	**150**	**7,649,779,782**	**0.00%**	**7,667,833,856**	**0.24%**	**7,725,891,803**	**0.99%**
	5	5	19	7,486,046,509	150	7,486,046,509	0.00%	7,486,046,509	0.00%	7,486,046,509	0.00%
	6	**4**	**1111**	**7,285,063,037**	**150**	**7,285,063,037**	**0.00%**	**7,374,490,986**	**1.23%**	**7,449,730,787**	**2.26%**
	6	**5**	**1536**	**7,180,559,304**	**150**	**7,180,559,304**	**0.00%**	**7,200,322,284**	**0.28%**	**7,279,374,203**	**1.38%**
	6	6	23	7,071,536,179	150	7,071,536,179	0.00%	7,071,536,179	0.00%	7,071,536,179	0.00%
(0.8,0.8)	3	1	136	9,923,897,797	150	9,923,897,797	0.00%	9,923,897,797	0.00%	9,923,897,797	0.00%
	3	2	2133	9,923,897,797	150	9,923,897,797	0.00%	9,923,897,797	0.00%	9,923,897,797	0.00%
	3	3	25	9,896,424,156	150	9,896,424,156	0.00%	9,896,424,156	0.00%	9,896,424,156	0.00%
	4	**2**	**3115**	**9,528,786,908**	**150**	**9,528,786,908**	**0.00%**	**9,540,071,444**	**0.12%**	**9,585,209,587**	**0.59%**
	4	**3**	**9416**	**9,406,173,571**	**150**	**9,406,173,571**	**0.00%**	**9,419,870,073**	**0.15%**	**9,474,656,082**	**0.73%**
	4	4	44	9,288,636,845	150	9,288,636,845	0.00%	9,288,636,845	0.00%	9,288,636,845	0.00%
	5	**3**	**7571**	**9,098,003,487**	**150**	**9,098,003,487**	**0.00%**	**9,121,088,609**	**0.25%**	**9,213,429,096**	**1.27%**
	5	**4**	**10431**	**8,962,997,030**	**150**	**8,962,997,030**	**0.00%**	**8,975,309,011**	**0.14%**	**9,016,780,504**	**0.60%**
	5	5	122	8,831,244,506	150	8,831,244,506	0.00%	8,831,244,506	0.00%	8,831,244,506	0.00%
	6	**4**	**7865**	**8,689,594,212**	**150**	**8,689,594,212**	**0.00%**	**8,718,293,872**	**0.33%**	**8,765,287,060**	**0.87%**
	6	**5**	**9567**	**8,562,974,155**	**150**	**8,562,974,155**	**0.00%**	**8,575,495,329**	**0.15%**	**8,625,580,026**	**0.73%**
	6	6	130	8,463,112,374	150	8,463,112,374	0.00%	8,463,112,374	0.00%	8,463,112,374	0.00%

Table 5. Results of CPLEX and the proposed SA for $(\alpha_C, \alpha_H) = (0.9, 0.9)$.

p	p_0	CPLEX Time (s)	CPLEX Optimal	SA Time (s)	Best Fitness	Best Gap%	Average Fitness	Average Gap%	Worst Fitness	Worst Gap%
3	1	610	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
3	2	3755	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
3	3	37	10,415,721,266	150	10,415,721,266	0.00%	10,415,721,266	0.00%	10,415,721,266	0.00%
4	2	5040	10,079,860,428	150	10,079,860,428	0.00%	10,079,860,428	0.00%	10,079,860,428	0.00%
4	**3**	**18865**	**10,023,330,347**	**150**	**10,023,330,347**	**0.00%**	**10,033,450,597**	**0.10%**	**10,069,507,134**	**0.46%**
4	**4**	**214**	**9,914,406,443**	**150**	**9,914,406,443**	**0.00%**	**9,916,452,779**	**0.02%**	**9,924,638,120**	**0.10%**
5	3	7500	9,683,085,222	150	9,683,085,222	0.00%	9,683,085,222	0.00%	9,683,085,222	0.00%
5	**4**	**35263**	**9,603,674,283**	**150**	**9,603,674,283**	**0.00%**	**9,605,204,001**	**0.02%**	**9,611,288,909**	**0.08%**
5	5	236	9,483,356,817	150	9,483,356,817	0.00%	9,483,356,817	0.00%	9,483,356,817	0.00%
6	**4**	**11800**	**9,342,665,140**	**150**	**9,342,665,140**	**0.00%**	**9,402,134,117**	**0.64%**	**9,499,766,190**	**1.68%**
6	**5**	**14571**	**9,228,121,701**	**150**	**9,228,121,701**	**0.00%**	**9,234,798,481**	**0.07%**	**9,253,722,274**	**0.28%**
6	6	214	9,114,839,991	150	9,114,839,991	0.00%	9,114,839,991	0.00%	9,114,839,991	0.00%

Table 6. Results of CPLEX and the proposed SA for $(\alpha_C, \alpha_H) = (0.6, 0.9)$ and $(\alpha_C, \alpha_H) = (0.8, 0.9)$.

(α_C, α_H)	p	p_0	CPLEX Time (s)	CPLEX Optimal	SA Time (s)	Best Fitness	Best Gap%	Average Fitness	Average Gap%	Worst Fitness	Worst Gap%
(0.6,0.9)	3	1	300	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
	3	**2**	**664**	**9,464,597,766**	**150**	**9,464,597,766**	**0.00%**	**9,469,114,939**	**0.05%**	**9,487,183,632**	**0.24%**
	3	3	10	8,826,647,392	150	8,826,647,392	0.00%	8,826,647,392	0.00%	8,826,647,392	0.00%
	4	**2**	**1470**	**9,311,789,331**	**150**	**9,311,789,331**	**0.00%**	**9,328,089,278**	**0.18%**	**9,371,717,517**	**0.64%**
	4	3	2379	8,606,860,144	150	8,606,860,144	0.00%	8,606,860,144	0.00%	8,606,860,144	0.00%
	4	4	13	8,020,821,500	150	8,020,821,500	0.00%	8,020,821,500	0.00%	8,020,821,500	0.00%
	5	3	3447	8,454,051,709	150	8,454,051,709	0.00%	8,454,051,709	0.00%	8,454,051,709	0.00%
	5	**4**	**2184**	**7,931,288,504**	**150**	**7,931,288,504**	**0.00%**	**7,935,357,216**	**0.05%**	**7,951,632,063**	**0.26%**
	5	5	19	7,486,046,509	150	7,486,046,509	0.00%	7,486,046,509	0.00%	7,486,046,509	0.00%
	6	**4**	**3870**	**7,862,099,067**	**150**	**7,865,114,631**	**0.04%**	**7,887,883,330**	**0.33%**	**7,914,894,574**	**0.67%**
	6	**5**	**2060**	**7,399,297,863**	**150**	**7,399,297,863**	**0.00%**	**7,405,072,805**	**0.08%**	**7,428,172,575**	**0.39%**
	6	6	23	7,071,536,179	150	7,071,536,179	0.00%	7,071,536,179	0.00%	7,071,536,179	0.00%
(0.8,0.9)	3	1	197	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
	3	2	1891	10,114,622,268	150	10,114,622,268	0.00%	10,114,622,268	0.00%	10,114,622,268	0.00%
	3	3	37	9,896,424,156	150	9,896,424,156	0.00%	9,896,424,156	0.00%	9,896,424,156	0.00%
	4	**2**	**13436**	**9,946,414,639**	**150**	**9,946,414,639**	**0.00%**	**9,949,126,631**	**0.03%**	**9,959,974,598**	**0.14%**
	4	**3**	**8661**	**9,618,082,826**	**150**	**9,618,082,826**	**0.00%**	**9,626,559,144**	**0.09%**	**9,660,464,415**	**0.44%**
	4	4	42	9,288,636,845	150	9,288,636,845	0.00%	9,288,636,845	0.00%	9,288,636,845	0.00%
	5	**3**	**14877**	**9,465,274,391**	**150**	**9,465,274,391**	**0.00%**	**9,466,439,129**	**0.01%**	**9,471,098,083**	**0.06%**
	5	4	6584	9,095,608,117	150	9,095,608,117	0.00%	9,095,608,117	0.00%	9,095,608,117	0.00%
	5	5	121	8,831,244,506	150	8,831,244,506	0.00%	8,831,244,506	0.00%	8,831,244,506	0.00%
	6	**4**	**7361**	**8,974,808,838**	**150**	**8,974,808,838**	**0.00%**	**9,024,554,256**	**0.55%**	**9,103,018,848**	**1.43%**
	6	**5**	**4819**	**8,666,718,166**	**150**	**8,666,718,166**	**0.00%**	**8,667,204,452**	**0.01%**	**8,669,149,594**	**0.03%**
	6	6	121	8,463,112,374	150	8,463,112,374	0.00%	8,463,112,374	0.00%	8,463,112,374	0.00%

Table 7. Results of CPLEX and the proposed ILS for $(\alpha_C, \alpha_H) = (0.6, 0.6)$ and $(\alpha_C, \alpha_H) = (0.8, 0.8)$.

(α_C, α_H)	p	p_0	CPLEX		ILS							
			Time (s)	Optimal	Time (s)	Best		Average		Worst	
						Fitness	Gap%	Fitness	Gap%	Fitness	Gap%
(0.6,0.6)	3	1	114	8,840,639,236	150	8,840,639,236	0.00%	8,840,639,236	0.00%	8,840,639,236	0.00%
	3	2	812	8,840,639,236	150	8,840,639,236	0.00%	8,840,639,236	0.00%	8,840,639,236	0.00%
	3	3	10	8,826,647,392	150	8,826,647,392	0.00%	8,826,647,392	0.00%	8,826,647,392	0.00%
	4	**2**	**2296**	**8,366,925,044**	**150**	**8,366,925,044**	**0.00%**	**8,390,963,442**	**0.29%**	**8,461,844,174**	**1.13%**
	4	3	1340	8,105,406,729	150	8,105,406,729	0.00%	8,105,406,729	0.00%	8,105,406,729	0.00%
	4	4	13	8,020,821,500	150	8,020,821,500	0.00%	8,020,821,500	0.00%	8,020,821,500	0.00%
	5	**3**	**4505**	**7,792,523,096**	**150**	**7,792,523,096**	**0.00%**	**7,798,381,805**	**0.08%**	**7,821,816,640**	**0.38%**
	5	**4**	**3576**	**7,649,779,782**	**150**	**7,649,779,782**	**0.00%**	**7,686,256,990**	**0.48%**	**7,739,946,097**	**1.18%**
	5	5	19	7,486,046,509	150	7,486,046,509	0.00%	7,486,046,509	0.00%	7,486,046,509	0.00%
	6	**4**	**1111**	**7,285,063,037**	**150**	**7,285,063,037**	**0.00%**	**7,328,293,970**	**0.59%**	**7,417,499,979**	**1.82%**
	6	5	1536	7,180,559,304	150	7,180,559,304	0.00%	7,180,559,304	0.00%	7,180,559,304	0.00%
	6	6	23	7,071,536,179	150	7,071,536,179	0.00%	7,071,536,179	0.00%	7,071,536,179	0.00%
(0.8,0.8)	3	1	136	9,923,897,797	150	9,923,897,797	0.00%	9,923,897,797	0.00%	9,923,897,797	0.00%
	3	2	2133	9,923,897,797	150	9,923,897,797	0.00%	9,923,897,797	0.00%	9,923,897,797	0.00%
	3	3	25	9,896,424,156	150	9,896,424,156	0.00%	9,896,424,156	0.00%	9,896,424,156	0.00%
	4	**2**	**3115**	**9,528,786,908**	**150**	**9,532,753,177**	**0.04%**	**9,557,281,527**	**0.30%**	**9,568,457,101**	**0.42%**
	4	**3**	**9416**	**9,406,173,571**	**150**	**9,406,173,571**	**0.00%**	**9,407,180,732**	**0.01%**	**9,411,209,373**	**0.05%**
	4	4	44	9,288,636,845	150	9,288,636,845	0.00%	9,288,636,845	0.00%	9,288,636,845	0.00%
	5	**3**	**7571**	**9,098,003,487**	**150**	**9,098,003,487**	**0.00%**	**9,108,566,558**	**0.12%**	**9,146,852,572**	**0.54%**
	5	**4**	**10431**	**8,962,997,030**	**150**	**8,962,997,030**	**0.00%**	**8,970,632,700**	**0.09%**	**8,993,398,947**	**0.34%**
	5	5	122	8,831,244,506	150	8,831,244,506	0.00%	8,831,244,506	0.00%	8,831,244,506	0.00%
	6	**4**	**7865**	**8,689,594,212**	**150**	**8,689,594,212**	**0.00%**	**8,704,732,781**	**0.17%**	**8,765,287,060**	**0.87%**
	6	**5**	**9567**	**8,562,974,155**	**150**	**8,562,974,155**	**0.00%**	**8,567,464,853**	**0.05%**	**8,581,461,377**	**0.22%**
	6	6	130	8,463,112,374	150	8,463,112,374	0.00%	8,463,112,374	0.00%	8,463,112,374	0.00%

Table 8. Results of CPLEX and the proposed ILS for $(\alpha_C, \alpha_H) = (0.9, 0.9)$.

p	p_0	CPLEX		ILS						
		Time (s)	Optimal	Time (s)	Best		Average		Worst	
					Fitness	Gap%	Fitness	Gap%	Fitness	Gap%
3	1	610	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
3	2	3755	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
3	3	37	10,415,721,266	150	10,415,721,266	0.00%	10,415,721,266	0.00%	10,415,721,266	0.00%
4	**2**	**5040**	**10,079,860,428**	**150**	**10,079,860,428**	**0.00%**	**10,136,593,764**	0.56%	**10,273,266,125**	**1.92%**
4	**3**	**18865**	**10,023,330,347**	**150**	**10,023,330,347**	**0.00%**	**10,031,390,665**	**0.08%**	**10,051,298,961**	**0.28%**
4	4	214	9,914,406,443	150	9,914,406,443	0.00%	9,914,406,443	0.00%	9,914,406,443	0.00%
5	**3**	**7500**	**9,683,085,222**	**150**	**9,683,085,222**	**0.00%**	**9,735,714,220**	**0.54%**	**9,814,252,716**	**1.35%**
5	4	35263	9,603,674,283	150	9,603,674,283	0.00%	9,603,687,870	0.00%	9,603,708,250	0.00%
5	5	236	9,483,356,817	150	9,483,356,817	0.00%	9,483,356,817	0.00%	9,483,356,817	0.00%
6	**4**	**11800**	**9,342,665,140**	**150**	**9,342,665,140**	**0.00%**	**9,417,131,064**	**0.80%**	**9,505,378,586**	**1.74%**
6	**5**	**14571**	**9,228,121,701**	**150**	**9,228,121,701**	**0.00%**	**9,235,477,574**	**0.08%**	**9,264,901,066**	**0.40%**
6	6	214	9,114,839,991	150	9,114,839,991	0.00%	9,114,839,991	0.00%	9,114,839,991	0.00%

Table 9. Results of CPLEX and the proposed ILS for $(\alpha_C, \alpha_H) = (0.6, 0.9)$ and $(\alpha_C, \alpha_H) = (0.8, 0.9)$.

(α_C, α_H)	p	p_0	CPLEX		ILS						
			Time (s)	Optimal	Time (s)	Best		Average		Worst	
						Fitness	Gap%	Fitness	Gap%	Fitness	Gap%
(0.6,0.9)	3	1	300	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
	3	2	664	9,464,597,766	150	9,464,597,766	0.00%	9,464,597,766	0.00%	9,464,597,766	0.00%
	3	3	10	8,826,647,392	150	8,826,647,392	0.00%	8,826,647,392	0.00%	8,826,647,392	0.00%
	4	**2**	**1470**	**9,311,789,331**	**150**	**9,311,789,331**	**0.00%**	**9,319,109,699**	**0.08%**	**9,348,391,171**	**0.39%**
	4	3	2379	8,606,860,144	150	8,606,860,144	0.00%	8,606,860,144	0.00%	8,606,860,144	0.00%
	4	4	13	8,020,821,500	150	8,020,821,500	0.00%	8,020,821,500	0.00%	8,020,821,500	0.00%
	5	3	3447	8,454,051,709	150	8,454,051,709	0.00%	8,454,051,709	0.00%	8,454,051,709	0.00%
	5	**4**	**2184**	**7,931,288,504**	**150**	**7,931,288,504**	**0.00%**	**7,935,357,216**	**0.05%**	**7,951,632,063**	**0.26%**
	5	5	19	7,486,046,509	150	7,486,046,509	0.00%	7,486,046,509	0.00%	7,486,046,509	0.00%
	6	**4**	**3870**	**7,862,099,067**	**150**	**7,872,289,169**	**0.13%**	**7,887,780,494**	**0.33%**	**7,907,205,858**	**0.57%**
	6	5	2060	7,399,297,863	150	7,399,297,863	0.00%	7,399,297,863	0.00%	7,399,297,863	0.00%
	6	6	23	7,071,536,179	150	7,071,536,179	0.00%	7,071,536,179	0.00%	7,071,536,179	0.00%
(0.8,0.9)	3	1	197	10,426,074,560	150	10,426,074,560	0.00%	10,426,074,560	0.00%	10,426,074,560	0.00%
	3	2	1891	10,114,622,268	150	10,114,622,268	0.00%	10,114,622,268	0.00%	10,114,622,268	0.00%
	3	3	37	9,896,424,156	150	9,896,424,156	0.00%	9,896,424,156	0.00%	9,896,424,156	0.00%
	4	**2**	**13436**	**9,946,414,639**	**150**	**9,946,414,639**	**0.00%**	**9,955,654,155**	**0.09%**	**9,961,813,833**	**0.15%**
	4	**3**	**8661**	**9,618,082,826**	**150**	**9,618,082,826**	**0.00%**	**9,620,078,674**	**0.02%**	**9,628,062,065**	**0.10%**
	4	4	42	9,288,636,845	150	9,288,636,845	0.00%	9,288,636,845	0.00%	9,288,636,845	0.00%
	5	**3**	**14877**	**9,465,274,391**	**150**	**9,465,274,391**	**0.00%**	**9,475,217,452**	**0.11%**	**9,495,150,032**	**0.32%**
	5	**4**	**6584**	**9,095,608,117**	**150**	**9,095,608,117**	**0.00%**	**9,122,870,647**	**0.30%**	**9,231,920,769**	**1.50%**
	5	5	121	8,831,244,506	150	8,831,244,506	0.00%	8,831,244,506	0.00%	8,831,244,506	0.00%
	6	4	7361	8,974,808,838	150	8,974,808,838	0.00%	8,974,808,838	0.00%	8,974,808,838	0.00%
	6	5	4819	8,666,718,166	150	8,666,718,166	0.00%	8,666,786,710	0.00%	8,667,060,888	0.00%
	6	6	121	8,463,112,374	150	8,463,112,374	0.00%	8,463,112,374	0.00%	8,463,112,374	0.00%

note that all the times in these tables are reported in seconds. Since there is no competing non-exact solution to our proposed solution approaches, test problems were solved using three approaches. First, each problem was solved using the CPLEX. Then, solutions were compared against the results of the proposed SA and ILS algorithms. For those cases in which the optimal solution was reached in all five runs, the whole row is highlighted in Tables 4 to 9. Results give clear indication that SA and ILS reach the optimal solutions for all five runs in 34 and 38 out of 60 instances, respectively. This finding is a clear indication that ILS dominates SA narrowly in reaching optimal solutions. Furthermore, in 54 out of 60 runs, the worst performance of the proposed SA has an error less than one percent, which equals 53 for ILS. Apparently, there is not a considerable difference between the two procedures considering their worst-case performance. As can be seen, in both of the proposed procedures, for no fewer than 58 out of 60 instances solved, the best error equals zero which apparently shows the promising performance of the proposed SA and ILS procedures.

On a close examination of results, one may find that there is some evidence that for test problems in which there is more difference between p and p_0, the errors of the proposed SA are higher. This could be directly attributed to the fact that for cases where p and p_0 take closer values, the problem becomes more similar to the classical p-hub median problem, which is clearly easier to solve.

Moreover, results show that both proposed procedures are superb in escaping local optima with the multi-start mechanism, which is activated after getting into local optima for a consecutive number of iterations. Figure 7 depicts the evolution of the objective function for the two procedures, with respect to the runtime for one of the problem instances. It is clear that although both algorithms reach negligible errors, SA reaches the optimal solution quicker. The same behavior is observable for almost all the instances solved. Therefore, SA seems a better procedure to be embedded in

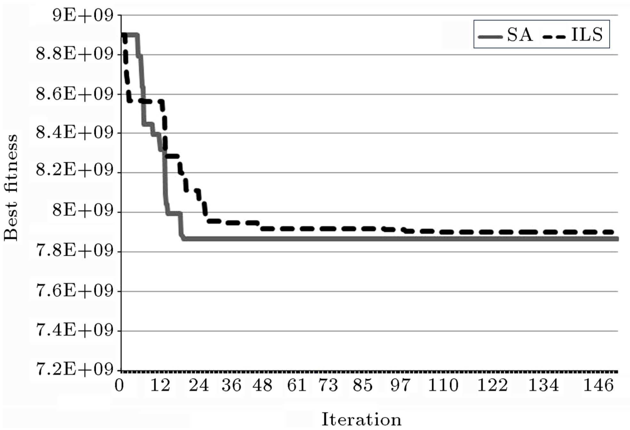

Figure 7. The trend of changing the optimal fitness throughout the algorithm for SA and ILS for $(\alpha_C, \alpha_H, p, p_0) = (0.6, 0.9, 6, 4)$.

simulation algorithms, considering concern about their runtimes.

At this point, we turn to the runtimes of the solution procedure. As already stated, the proposed procedures terminate after searching for 150 seconds. The outputs of the proposed SA and ILS show that regarding runtime, there are significant improvements compared to using CPLEX. In other words, the runtimes to reach the final solution are lower than CPLEX by more than two-hundred orders of magnitude. Moreover, in some cases, CPLEX is unable to reach optimal solutions in about ten hours. Furthermore, in the majority of cases, the proposed procedures reach optimal solutions in much less times than 150 seconds. Therefore, from a runtime point of view, the proposed procedures perform very well, and their benefits outweigh the drawback of reaching non-optimal solutions, in some cases.

Figure 8 compares the best, average, and worst performances of the proposed SA and ILS procedures, on average. Results clearly show that although the two algorithms perform nearly the same, better results are obtainable using ILS.

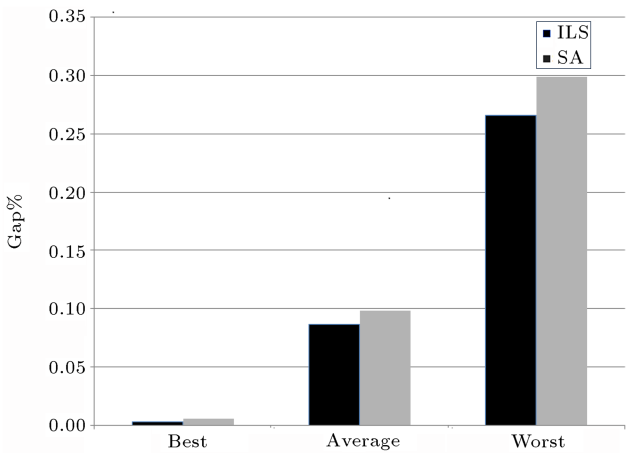

Figure 8. The mean of worst, average, and best performances of SA and ILS.

Overall, it is to be concluded that for the time-restricted run of the algorithms, SA outperforms ILS, since it reaches optimality quicker. However, in cases where time is not a measure of interest, ILS performs slightly better in reaching optimal solutions. In addition, both the proposed procedures perform well in reaching optimal solutions.

For further analysis of the results, we targeted the *Turkish Network* dataset, with the same assumptions as in Yaman [5]. It has been assumed that there are 81 potential cities in Turkey to establish hub nodes. Two of these nodes (Istanbul and Ankara) are fixed to host central hubs and the experiments were performed with different values of $p = 7, ..., 12$. However, we limited the runtime to 500 seconds, based on the difference in problem size. Results of running our proposed algorithm for different values of p show that both SA and ILS perform well in improving the optimal solution and they do not get stuck in local optima throughout the solution time. However, relaxing the constraint of fixing the locations of central hubs will bring about difficulties that are not addressed in this paper. Figure 9 shows how these two procedures perform on a problem with $p = 9$ and $p_0 = 2$. Results show that the average errors of SA and ILS for the problem with $p = 9$ and $p_0 = 2$ are %1.64 and %1.88, respectively.

6. Conclusion and future research areas

To bring this paper to a close, in this section, a brief conclusion is given and some avenues for further research are proposed. In this paper, two metaheuristics are proposed to solve the single-allocation hierarchical

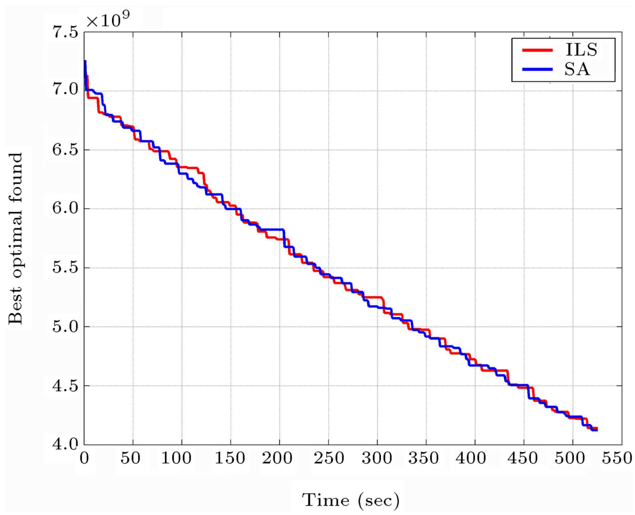

Figure 9. Comparing the performance of SA and ILS for the Turkish dataset (the values of the x axis are reported in seconds.)

hub location problem. A heuristic to initiate the solution algorithm is proposed which performs well in providing the solution procedure with high-quality initial solutions. Statistical tests on the problem parameters were conducted, which clearly showed the role of selecting parameter levels in reaching results. The results of using the CAB dataset showed that the solution procedures are able to reach solutions in less time and with a very high level of precision. Results showed that although there is no significant difference between SA and ILS, SA could be regarded as the better procedure to solve SA-H-MP in time-restricted cases. However, ILS outperforms SA in reaching optimality for longer runtimes.

There are some possibilities for future research studies. First of all, we believe that this paper is just a starting point in using metaheuristics in hierarchical location problems. This means that other heuristics and metaheuristics deserve to be studied to solve SA-H-MP. Moreover, since the problem is still rather untouched, many variants could still be studied. An appealing future extension could be to consider the covering and center variants of the SA-H-MP. Another interesting future research could be inclusion of capacities in the model. Moreover, there is the possibility of using uncertain parameters in the problem, such as the flow matrix or even the discount factor. Last, but not least, one may pursue this research using a fuzzy goal programming approach, which is a practical way of designing networks (see [55]). We strongly believe that these extensions can greatly enrich the literature and deserve to be considered in the future.

References

1. Rezaei, M. and Zarandi, M.H.F. "Facility location via fuzzy modeling and simulation", *Applied Soft Computing*, **11**(8), pp. 5330-5340 (2011).
2. Klose, A. and Drexl, A. "Facility location models for distribution system design", *European Journal of Operational Research*, **162**(1), pp. 4-29 (2005).
3. Hamacher, H.W. and Meyer, T. "New developments on hub location", *Computers & Operations Research*, **36**(12), pp. 3087-3087 (2009).
4. Alumur, S. and Kara, B.Y. "Network hub location problems: The state of the art", *European Journal of Operational Research*, **190**(1), pp. 1-21 (2008).
5. Yaman, H. "The hierarchical hub median problem with single assignment", *Transportation Research Part B: Methodological*, **43**(6), pp. 643-658 (2009).
6. Goldman, A.J. "Optimal location for centers in a network", *Transportation Science*, **3**, pp. 352-360 (1969).
7. O'Kelly, M.E. "A quadratic integer program for the location of interacting hub facilities", *European Journal of Operational Research*, **32**(3), pp. 393-404 (1987).
8. Campbell, J.F. "Integer programming formulations of discrete hub location problems", *European Journal of Operational Research*, **72**(2), pp. 387-405 (1994).
9. Skorin-Kapov, D., Skorin-Kapov, J. and O'Kelly, M. "Tight linear programming relaxations of uncapacitated p-hub median problems", *European Journal of Operational Research*, **94**(3), pp. 582-593 (1996).
10. Ebery, J. "Solving large single allocation p-hub problems with two or three hubs", *European Journal of Operational Research*, **128**(2), pp. 447-458 (2001).
11. Sahin, G. and Sural, H. "A review of hierarchical facility location models", *Computers & Operations Resaerch*, **34**(8), pp. 2310-2331 (2007).
12. Alumur, S., Yaman, H. and Kara, B.Y. "Hierarchical multimodal hub location problem with time-deffinite deliveries", *Transportation Research Part E*, **48**, pp. 1107-1120 (2012).
13. Lee, J.M. and Lee, Y.H. "Tabu based heuristics for the generalized hierarchical covering location problem", *Computers & Industrial Engineering*, 58, pp. 638-645 (2010).
14. Davari, S. and Fazel Zarandi, M.H. "The single-allocation hierarchical hub median location problem with fuzzy demands", *African Journal of Business Management*, **6**(1), pp. 347-360 (2012).
15. Calik, H., Alumur, S.A., Kara, B.Y. and Karasan, O.E. "A tabu-search based heuristic for the hub covering problem over incomplete hub networks", *Computers & Operations Research*, **36**(12), pp. 3088-3096 (2009).
16. Cunha, C.B. and Silva, M.R. "A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil", *European Journal of Operational Research*, **179**(3), pp. 747-758 (2007).
17. Skorin-Kapov, D. and Skorin-Kapov, J. "On tabu search for the location of interacting hub facilities", *European Journal of Operational Research*, **73**(3), pp. 502-509 (1994).
18. Meyer, T., Ernst, T. and Krishnamoorthy, M. "A 2-phase algorithm for solving the single allocation p-hub center problem", *Computers & Operations Research*, **36**(12), pp. 3143-3151 (2009).
19. Abdinnour-Helm, S. "Using simulated annealing to solve the p-hub median problem", *International Journal of Physical Distribution & Logistics Management*, **31**(3), pp. 203-220 (2001).
20. Ilic, A., Urosevic, D., Brimberg, J. and Mladenovic, N. "A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem", *European Journal of Operational Research*, **206**(2), pp. 289-300 (2010).
21. Eiselt, H.A. and Marianov, V. "A conditional p-hub location problem with attraction functions", *Computers & Operations Research*, **36**(12), pp. 3128-3135 (2009).
22. Gavriliouk, E.O. "Aggregation in hub location problems", *Computers & Operations Research*, **36**(12), pp. 3136-3142 (2009).

23. Sim, T., Lowe, T.J. and Thomas, B.W. "The stochastic p-hub center problem with service-level constraints", *Computers & Operations Research*, **36**(12), pp. 3166-3177 (2009).

24. Kim, H. and O'Kelly, M.E. "Reliable p-hub location problems in telecommunication Networks", *Geographical Analysis*, **41**, pp. 283-306 (2009).

25. Campbell, J.F. "Hub location for time definite transportation", *Computers & Operations Research*, **36**(12), pp. 3107-3116 (2009).

26. Lin, C.C. and Lee. S.C. "The competition game on hub network design", *Transportation Research Part B: Methodological*, **44**(4), pp. 618-629 (2010).

27. Correia, I., Nickel, S. and Saldanha-da-Gama, F. "Single-assignment hub location problems with multiple capacity levels", *Transportation Research Part B: Methodological*, **44**(8-9), pp. 1047-1066 (2010).

28. Lin, C.C. "The integrated secondary route network design model in the hierarchical hub-and-spoke network for dual express services", *International Journal of Production Economics*, **123**(1), pp. 20-30 (2010).

29. Vasconcelos, A.D., Nassi, C.D. and Lopes, L.A.S. "The uncapacitated hub location problem in networks under decentralized management", *Computers & Operations Research*, **38**(12), pp. 1656-1666 (2011).

30. Correia, I., Nickel, S. and Saldanha-da-Gama, F. "Hub and spoke network design with single-assignment, capacity decisions and balancing requirements", *Applied Mathematical Modelling*, **35**(10), pp. 4841-4851 (2011).

31. Contreras, I., Cordeau, J.F., and Laporte, G. "Stochastic uncapacitated hub location", *European Journal of Operational Research*, **212**(3), pp. 518-528 (2011).

32. Catanzaro, D., Gourdin, E., Labbe, M. and Ozsoy, F.A. "A branch-and-cut algorithm for the partitioning-hub location-routing problem", *Computers & Operations Research*, **38**(2), pp. 539-549 (2011).

33. Yaman, H. "Allocation strategies in hub networks", *European Journal of Operational Research*, **211**(3), pp. 442-451 (2011).

34. Puerto, J., Ramos, A.B. and Rodriguez-Chia, A.M. "Single-allocation ordered median hub location problems", *Computers & Operations Research*, **38**(2), pp. 559-570 (2011).

35. Fazel Zarandi, M.H., Davari, S. and Haddad Sisakht, S.A. "The Q-coverage multiple allocation hub covering problem with mandatory dispersion", *Scientia Iranica E*, **19**(3), pp. 902-911 (2012).

36. Davari, S., Fazel Zarandi, M.H. and Turksen, I.B. "The incomplete hub-covering location problem considering imprecise location of demands", *Scientia Iranica E*, **20**(3), pp. 983-991 (2013).

37. Luer-Villaga, A. and Marianov, V. "A competitive hub location and pricing problem", *European Journal of Operational Research*, **231**, pp.734-744 (2013).

38. De Camargo, R.C., De Miranda, G. and Lokketangen, A. "A new formulation and an exact approach for the many-to-many hub location-routing problem", *Applied Mathemtical Modelling*, **37**, pp. 7465-7480 (2013).

39. Kirkpatrick, S., Gelatt, C.D. and Vecchi, M.P. "Optimization by simulated annealing", *Science*, **220**, pp. 671-680 (1983).

40. Fazel Zarandi, M.H., Hemmati, A., Davari, S. and Turksen, I.B. "Capacitated location-routing problem with time windows under uncertainty", *Knowledge-Based Systems*, **37**, pp. 480-489 (2013).

41. Zhang, R. and Wu, C. "A hybrid immune simulated annealing algorithm for the job shop scheduling problem", *Applied Soft Computing*, **10**(1), pp. 79-89 (2010).

42. Karasulu, B. and Korukoglu, S. "A simulated annealing-based optimal threshold determining method in edge-based segmentation of grayscale images", *Applied Soft Computing*, **11**(2), pp. 2246-2259 (2011).

43. Rodríguez, V., Alvarez, M.J. and Barcos, L. "Hub location under capacity constraints", *Transportation Research Part E: Logistics and Transportation Review*, **43**(5), pp. 495-505 (2007).

44. Rodríguez-Martín, I. and Salazar-González, J.J. "Solving a capacitated hub location problem", *European Journal of Operational Research*, **184**(2), pp. 468-479 (2008).

45. Ernst, A.T. and Krishnamoorthy, M. "Efficient algorithms for the uncapacitated single allocation p-hub median problem", *Location Science*, **4**(3), pp. 139-154 (1996).

46. Mohammadi, M., Tavakkoli-Moghaddam, R., Tolouei, H. and Yousefi, M. "Solving a hub covering location problem under capacity constraints by a hybrid algorithm", *Journal of Applied Operational Research*, **2**(2), pp. 109-116 (2010).

47. Ernst, A. and Krishnamoorthy, M. "Solution algorithms for the capacitated single allocation hub location problem", *Annals of Operations Research*, **86**, pp. 141-159 (1999).

48. Chen, J.F. "A hybrid heuristic for the uncapacitated single allocation hub location problem", *Omega*, **35**(2), pp. 211-220 (2007).

49. Lundy, M. and Mees, A. "Convergence of an annealing algorithm", *Mathematical Programming*, **34**, pp. 111-124 (1986).

50. Dong, X., Huang, X. and Chen, P. "An iterated local search algorithm for the permutation flowshop problem with total flowtime criterion", *Computers & Operations Research*, **36**(5), pp. 1664-1669 (2009).

51. Lourenco, H.R., Martin, O.C. and Stutzle, T. "Iterated local search", *Handbook of Metaheuristics*, F. Glover and G. Kochenberger, Kluwer, Boston, pp. 321-353 (2002).

52. Stützle, T. "Iterated local search for the quadratic assignment problem", *European Journal of Operational Research*, **174**(3), pp. 1519-1539 (2006).

53. Hashimoto, H., Yagiura, M. and Ibaraki, T. "An iterated local search algorithm for the time-dependent vehicle routing problem with time windows", *Discrete Optimization*, **5**(2), pp. 434-456 (2008).

54. Blum, C., Puchinger, J., Raidi, G.R. and Roli, A. "Hybrid metaheuristics in combinatorial optimization: A survey", *Applied Soft Computing*, **11**(6), pp. 4135-4151 (2011).

55. Fazel Zarandi, M.H., Davari, S. and Haddad Sisakht, S.A. "Design of a closed-loop supply chain (CLSC) model using an interactive fuzzy goal programming", *International Journal of Advanced Manufacturing Techology*, **56**(5-8), pp. 809-821 (2011).

Biographies

Mohammad Hossein Fazel Zarandi is Professor in the Department of Industrial Engineering at Amirkabir University of Technology, Tehran, Iran, and a member of the Knowledge-Information Systems Laboratory at the University of Toronto, Canada. His main research interests focus on intelligent information systems, soft computing, computational intelligence, fuzzy sets and systems, multi-agent systems, networks, metaheuristics, and optimization. Professor Fazel Zarandi has published numerous books, scientific papers and technical reports in the above areas, most of which are also accessible on the web. He has taught several courses in fuzzy systems engineering, decision support systems, management information systems, artificial intelligence and expert systems, systems analysis and design, scheduling, neural networks, simulations, and production planning and control, at several universities in Iran and North America.

Soheil Davari received his PhD degree in Industrial Engineering, in 2011, and is currently Postdoctoral Fellow at Sabanci University, Istanbul, Turkey. His research is focused on health care management systems, large-scale optimization, stochastic processes, and location theory.

Seyyed Ali Haddad Sisakht obtained his MS degree in Industrial Engineering from Sharif University of Technology, Tehran, Iran, and is currently a PhD degree student in Iowa State University, USA.